

Diverse Large-Scale ITS Dataset Created from Continuous Learning for Real-Time Vehicle Detection

Justin A. Eichel¹, Akshaya Mishra¹, Nicholas Miller², Nicholas Jankovic²,
Mohan A. Thomas, Tyler Abbott, Douglas Swanson, Joel Keller

Abstract—In traffic engineering, vehicle detectors are trained on limited datasets resulting in poor accuracy when deployed in real world applications. Annotating large-scale high quality datasets is challenging. Typically, these datasets have limited diversity; they do not reflect the real-world operating environment. There is a need for a large-scale, cloud based positive and negative mining (PNM) process and a large-scale learning and evaluation system for the application of traffic event detection. The proposed positive and negative mining process addresses the quality of crowd sourced ground truth data through machine learning review and human feedback mechanisms. The proposed learning and evaluation system uses a distributed cloud computing framework to handle data-scaling issues associated with large numbers of samples and a high-dimensional feature space. The system is trained using AdaBoost on 1,000,000 Haar-like features extracted from 70,000 annotated video frames. The trained real-time vehicle detector achieves an accuracy of at least 95% for 1/2 and about 78% for 19/20 of the time when tested on approximately 7,500,000 video frames. At the end of 2015, the dataset is expect to have over one billion annotated video frames.

Index Terms—sample selection, AdaBoost, positive mining, negative mining, real-time vehicle detection, Haar-like feature space, distributed learning and evaluation, large-scale traffic datasets.

I. INTRODUCTION

Automatic traffic event detection technologies play a major role in safe, reliable and efficient operations of road transportation systems [1], including traffic surveillance [2], vehicle presence detection [3], traffic density estimation, emergency response, traffic re-routing and real-time optimized signal control [1], [3]. Sensing, transmitting, and computing [4] are three major technological components of an automatic traffic event detector. Various sensors [5] including road-tubes, loop-detectors, radars and cameras are used to collect measurements. The measured data are analyzed using various data analytic methods to build effective traffic management systems. Although simple non-video-based sensors can provide a higher signal-to-noise ratio than video cameras, video-based traffic measurements systems are very popular for two reasons. First, the video-based detector signal can be reviewed by humans [6]. Second, advanced computer vision algorithms can be employed at different stages of data collections to

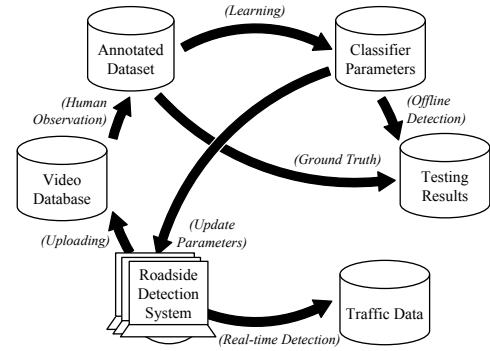


Fig. 1. A scalable platform for learning and evaluating a real-time vehicle detection system. A large network of connected cameras and real-time roadside processors captures and streams traffic video. A distributed machine learning system continually samples ground truth data and supervised training examples. It learns better classification parameters and evaluates real-time vehicle detection on a large testing set. Improvements are incorporated by sending incremental parameter updates to the roadside processors.

extract scalable information that can be used in designing efficient intelligent transportation systems (ITS) [1], [7], [8].

Many recent video-based vehicle detectors rely on machine learning to detect and classify vehicles [9]. The classifiers have a number of parameters that need to be trained to ensure that the detector correctly classifies objects of interest, such as vehicles or pedestrians, while correctly classifying ‘non-vehicles’, such as roadway or trees. When given one or more video frames, a video-based vehicle detector might try to localize vehicles using motion, shape, and appearance-based features. For example, a Haar feature vector [10] can be extracted locally for each region of interest in a video frame. Then, a classifier can transform the feature vector into a score, indicating how similar the region of interest is to a vehicle. The parameters controlling how the classifier transforms the feature vectors into binary or multiple class labels can be trained using samples of real traffic event data, namely images of vehicles and images of non-vehicles.

Many existing video detectors are trained from existing traffic datasets [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. However, many of these datasets are not actively growing, and they do not contain sufficient diversity to train, validate, and test a generalized real-time vehicle detector to be deployed in a real world application [21]. These datasets do not contain a sufficient number of diverse samples of

¹ Eichel and Mishra contributed equally

² Miller and Jankovic contributed equally
Miovision Technologies Inc.

Manuscript received October 15, 2014; revised December ?, 2014.

weather conditions, camera perspectives, roadway conditions, and roadway configurations. Collection of this data is generally cost prohibitive due to the quantity of annotations required to sample each scenario. Maintenance of the dataset adds additional cost and it may well be infeasible to continuously add under-sampled scenarios to the dataset. Partitioning a dataset into training and test is an important consideration. Detection algorithms are most effectively trained with diverse training sets with strong representation from all decision boundaries. Recent work [22] on weakly supervised classifier have shown how continuous active learning using positive and negative mining can accomplish this, substantially improving the performance of general object detectors.

Further, many existing vehicle learning and evaluation systems are not designed to efficiently process billions of traffic event annotations [8]. Storage is required to archive the dataset, and each annotation may need to be retrieved multiple times for each training session. As new samples are continuously added to the dataset, the video detector must be periodically retrained. Distributed computing systems [23], [24] are designed to address the storage and processing requirements. For example, Netflix stores 6.5 billion hours of video as of the first quarter of 2014 and makes extensive use of Amazon's Web Services (AWS) [25] for their transcoding processes. Sec. II-C1 provides background on AWS and other distributed computing environments upon which the proposed learning and evaluation system is built.

The authors propose three main contributions to ITS video detection:

- 1) a large-scale ITS positive and negative data mining process for training and validation sample selection,
- 2) a large-scale ITS learning and evaluation system, and
- 3) a large-scale ITS traffic event dataset, soon to be available for research collaboration.

The remainder of the paper details the background, the proposed contributions, experimental results, and conclusions. The background, Sec. II, describes the application, technology and terminology surrounding ITS vehicle detectors, (Sec. II-A, Sec. II-B), the scope and availability of existing ITS related datasets, and related machine learning methods (Sec. II-C). Sec. III presents the proposed traffic event positive and negative mining process (Sec. III-B) and the proposed learning and evaluation system (Sec. III-D), see Fig. 1. The third section (Sec. IV) presents experimental results illustrating the impact of training dataset composition on overall accuracy when evaluated on the entire testing dataset. Sec. V presents concluding remarks and discussion of future work with the intent of making the proposed ITS dataset available for research collaboration.

II. BACKGROUND

This section provides preliminary context with a discussion of detection technology and related techniques. The section begins with a summary of vehicle detection applications and technology alternatives. Second, a broad overview of various video-based vehicle detection technologies is provided. Third, existing ITS video datasets are considered. Finally, the section

concludes with a discussion of distributed computing environments and their application to the problem of training vehicle detection systems.

A. Vehicle detector technology

Road infrastructure is a significant investment made by governments and private agencies. The engineering of these traffic systems requires current usage data. Vehicle detector technologies are a key tool in collecting traffic data and extracting meaningful information from them for the purpose of building better traffic management systems. Various traffic studies, such as Automatic Traffic Recorder (ATR) studies, turning movement counts (TMCs), origin-destination (O-D) studies and travel time (TT) studies help in infrastructure budgeting, time of day traffic signal timing, vehicle density estimation for roadside advertising, toll route usage pricing calculation, and class based lane usage. Further, real-time vehicle detection technologies can help in traffic re-routing, and reduce wait times by performing demand-based signal control.

Since the introduction of inductive loops in the 1960s [26], many other intrusive sensors such as road tubes, piezoelectric cables, weigh-in-motion sensors, magnetoresistive sensors [3] and micro-ferromagnetic induction coil sensors [5] are being installed in or on the roadway to detect vehicles. Non-intrusive sensors are installed outside of the direct vehicle path, typically above the ground. Examples include microwave radar systems, infrared radar (Lidar) systems, passive infrared (PIR) sensors, ultrasonic sensors, acoustic sensors, video imaging vehicle detection systems [3] and optical beam break sensors [27]. Seismic sensors can also detect vehicles [28]; these can attach to the ground, usually beside the roadway. In addition to physical sensors, the digital era allows vehicle localization through the use of transponders, smartphones [29], Bluetooth devices [30] and vehicular ad-hoc networks [4].

B. Overview of vehicle detection through video

Given all of the available technologies, real-time video offers a visual source of vehicle and environment data and does not require the vehicle or its passenger to possess any specialized technology. Video traffic data is ideally suited to learning-based computer vision algorithms [31] and complements the data-driven intelligent transportation systems philosophy [1].

In general, a vehicle detector indicates the presence of a vehicle through the following mathematical process. Measurements, \vec{m} , in the form of pixel intensity, are obtained for each region of interest, r , within a video frame, I . To reduce complexity of the classifier, the detector takes \vec{m} and extracts a feature vector \vec{f} (also referred to as "the features"), which lies in a feature space. Ideally, given a set of annotated positive vehicle sightings S_p , and negative vehicle sightings S_n , each feature vector \vec{f}_p extracted from S_p will occupy a distinct region in the feature space, differentiated from the set of feature vectors \vec{f}_n extracted from S_n .

Existing video-based vehicle detector feature spaces typically fit into two categories, (a) motion-based and (b) appearance and geometry-based features. Motion-based features

allow the detector to identify moving vehicles from temporal changes over a set of consecutive video frames. Motion-based video detector algorithms are intuitive, easy to implement, computationally efficient, and can be implemented in real-time using low cost embedded systems. However, if I_{bg} does not account for dynamic changes due to illumination, glare, rain, snow, fog, wind, or other weather-related effects, the detector may fail to distinguish between dynamic background regions and moving vehicles [6], [21]. Further, the performance of motion-based detectors degrades significantly when the object of interest is stationary or moves slowly. The detector may also misclassify vehicles in the presence of stationary or independently moving motion fields, such as if the vehicle is occluded due to trees or overhead wires. As exemplified in Fig. 2, vehicle detectors using only motion features cannot accommodate the aforementioned real-world scenarios [21]. On the other hand, machine learning based vehicle detection using shape features has shown promising results for classifying internet images [32].

C. Machine learning based vehicle detection

The three main components of machine learning based detectors are feature extraction, feature selection and classifier design. Typically, feature extraction techniques estimate salient features, \vec{f} , based on a vehicle's appearance and shape. Such features can be calculated at various spatial scales, locally at a single pixel location, (i, j) , in I , regionally for a neighborhood, $\mathcal{N}_{i,j}$, surrounding (i, j) , or globally over all of I . Many appearance and geometry-based features include Haar-like features [10], histogram of oriented gradients (HOG) [33], scale-invariant feature transform (SIFT) [34], SURF [35], ORB [36], Gabor filters [37], and super pixels [38]. The selection of appropriate features varies by application depending on the properties of the class that will be detected. For example, Fig. 3 illustrates the SURF feature response applied to real-world ITS video produces strong responses for both vehicles and non-vehicles, while a classifier trained on Haar-like features [39], produces strong responses for vehicles and weaker responses for non-vehicles. The proposed learning and evaluation system in Sec. III is based on Haar-like features applied at various spatial scales.

Further, since each feature has a computational and memory cost and video detectors must operate on cost effective hardware in real-time, the process of feature selection is useful to determine which subset of features, from the set of all possible features, contribute the most to the video detector accuracy. Fortunately, the feature selection process can be performed offline using principal components analysis (PCA), independent component analysis (ICA) [40], and unsupervised clustering techniques [7]. Other feature selection techniques are built into classifier training. For example, support vector machines (SVM), bagging and boosting [41], and convolutional neural networks [32], recurrent neural networks and incremental recurrent neural networks [24] determine which features are significant as part of their training process. Computational performance is improved since only the selected features are calculated as the real-time vehicle detector evaluates each r in each I .

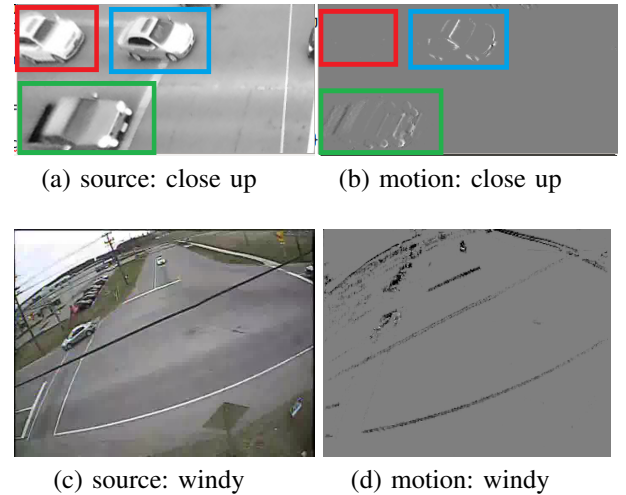


Fig. 2. Given (a) a roadway, Harrison's implementation of a Reichardt motion model [42], [43] creates (b) strong responses for the moving vehicles, (red and (blue), but fails to create any response for the stationary vehicle (green). Given (c) a different roadway location containing minor wind conditions, the motion response (d) is strong for moving background scenery, making it difficult to determine which motion responses correspond to background and which ones correspond to vehicles.

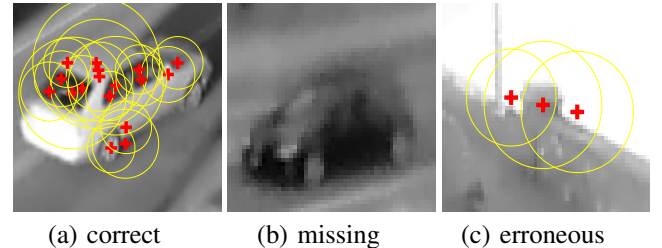


Fig. 3. SURF keypoints [35] detected using constant sensitivity parameters. Note that the car in (a) has keypoints which may be matched for detection, while the car in (b) has no SURF keypoints. Furthermore, a background object, a bridge, in (c) has keypoints, which could result in a false positive detection.

Once features are defined, the classifier is then trained such that a given \vec{f} is classified as either a vehicle or non-vehicle based on the how similar \vec{f} is to the collection of all \vec{f}_p or the collection of all \vec{f}_n . A simple classifier, such as nearest-neighbour [7], may assign \vec{f} as a vehicle if the distance of \vec{f} to the nearest example in \vec{f}_p is less than the distance of \vec{f} to the nearest \vec{f}_n sample. Based upon applications, classifier complexity can increase to produce advanced classifiers such as neural networks (NN), cascading classifiers (CC), boosted classifiers, and support vector machines (SVM). Boosting methods are derived from the idea that many simple classifiers can be combined to be more accurate than any one simple classifier. Freund et al. [44] detail how a collection of discriminants, each with a classification accuracy of at least 50%, can be combined into a single classifier with significantly higher classification accuracy [41], [45].

The real-time and real-world classification performance of machine learning based vehicle detectors significantly depends on the availability of a large-scale high quality annotated vehicle dataset and a scalable learning and evaluation system. The scale and scope of several major ITS datasets published between 1998 and 2014 are summarized in Table I. The

creation of a large high quality dataset requires a good positive and negative mining framework as well as dedicated resources to perform the review tasks [46].

1) *Distributed computing environments*: Given a large amount of data and a high dimensional feature space, one method of processing the data is to construct high performance computing (HPC) infrastructure through computer clusters [47]. These cluster infrastructures provide methods for sharing memory and distributing computations over a large number of computers. Although the process of building and configuring a cluster has been simplified [48], this solution requires capital investment and maintenance costs, which requires in-house technology experts. Further, the process of designing an algorithm and evaluating it on the dataset requires almost no demand during the design process, but high computational demand during the evaluation process; demand for HPC may be high, but variable, and the cluster has insufficient capacity to complete jobs in a timely manner. Elastic cloud computing offers an alternative option which can dynamically scale to match variable work loads, such as on-demand video transcoding [49]. For instance, Amazon Web Services (AWS) [50] offers distributed memory storage connected through high speed networks to on-demand computing systems. Human annotators can generate data from anywhere in the world and the learning and evaluation system can scale to accommodate large-scale datasets of annotated video. The authors propose a large-scale learning and evaluation system, built on top of AWS, that is capable of training and evaluating a vehicle traffic detector on billions of annotations.

D. Active and continuous learning

During the training phase, it is not uncommon [51], [7] to use manually partitioned datasets to test and train with some number of positive and negative samples. Tamersoy et al. [52] used ‘difficult’ negative samples, containing vehicle components, with the intention of training more robust detection algorithms. Such an approach is defensive; the training set is selected in anticipation of likely failure modes in the detection algorithm. Including ‘difficult’ samples in the training set can improve classifiers at otherwise ambiguous decision boundaries.

Sivaraman [8] demonstrated how an active learning approach can effectively be used to train a robust real-time on-road vehicle tracking algorithm. In their framework, two training iterations were used to focus on informative samples. The first iteration trains the detector with a manually partitioned data set. The trained detector is then evaluated with an independent test set. Results of this evaluation are selectively sampled to augment and prune the original training set in such a way that difficult decision boundaries are more heavily represented. The effect of this was to create a more robust detector, reducing the number of false positives. Positive and negative mining techniques are an important component of active and continuous learning because they reduce a large pool of datasets into a manageable and representative set [22]. A classifier trained on samples selected using positive and negative mining provides better classification accuracy compared

to a classifier trained on general samples [22]. Table I includes several datasets appropriate for training, and in some cases, testing video detection algorithms.

E. Summary of issues related to ITS vehicle detection systems

1) *ITS Datasets*: Existing datasets such as [18] and [51] include annotations to localize general vehicle objects, yet do not provide detail about scene conditions and operating environments. Although there are a few domain specific annotations such as vehicle class present, they are limited in scope. They may be limited in number of locations or perspectives. Very often, diverse weather conditions and lighting conditions are not present, and where they are, these conditions are not annotated. The datasets did not annotate roadways or intersections and, consequently, could not provide lane geometries or lane assignment of vehicles. Generally, training a detector against datasets with such limited scope does not provide confidence for real-world performance. Further, there is no information regarding the data acquisition and annotation process - a notable exception is Kasturi et al [51], which demonstrated the benefits of a formal annotation review process. However, in general, basic questions including, “are the videos coming from different sources?”, “was the annotation process audited for reliability?” [46], and “what are the conditions represented?” remain unanswered with most of the datasets.

2) *Learning and evaluation platform*: Carefully designed sample sets and feature vectors play a significant role in the success of machine learning based vehicle detection system [46]. Selecting an optimal set of training samples and features vectors that produce minimal generalization error, require a daunting amount work in designing a scalable computational frame work. Existing learning and evaluation platforms are unable to handle billions of samples and millions of features from which an optimal set of quality training samples and feature vectors can be selected. Continuous positive and negative mining tools have been effectively used for general purpose computer vision detection and tracking applications [53], but they have not yet been applied to large-scale ITS datasets.

III. PROPOSED SOLUTIONS

As discussed in Sec. II-C, to the best of the authors’ knowledge, a comprehensive ITS traffic dataset for vehicle detection does not exist; existing datasets do not contain sufficient diversity to train a vehicle detector for deployment in real-world conditions. As a result, the fundamental objective of this paper is to develop a large-scale diverse traffic dataset, through a data mining process for semi-supervision of learning algorithms, and a learning and evaluation platform, for estimating the optimal parameters of a given classifier and feature extraction method.

Fig. 4 illustrates the composition of a large-scale learning and evaluation system. The system contains data sensing, through real-world cameras, and a human annotation process that labels objects of interest in video frames. A training algorithm, based on dataset mining, is described in Sec. III-B,

TABLE I
ITS DATASETS

Year	Author	Num. Videos	Num. Annotations	Resolution	Color	Classes	Description
2000	Schneiderman [18]	213	213	various	gray	car	various still images
2000	Papageorgiou [15]	516	516	128 × 128	color	car	still images with head-on view
2001	Makris [19]	20	0	640 × 480	color	none	various parking lot views
2004	Agarwal [17]	1328	1,050	100 × 40	gray	car	278 testing images at various scale
2006	Bileschi [20]	3547	27,666	1280 × 960	color	many	still images of cars, pedestrians and more
2007	Saunier [14]	N/A	2941	N/A	gray	N/A	10 to 60-sec clips from common location
2007	Saunier [14]	1	47,084	N/A	N/A	N/A	1-hour segment at an intersection
2009	Kasturi [51]	100	~ 37,500	720 × 480	color	car	~ 2.5 min videos with annotated I-frames
2009	Xiaogang [13]	1	540	720 × 480	color	car,ped	1.5-hour segment at an intersection
2011	Patrick [12]	630	315	216 × 384	gray	car	optical flow data for each frame
2011	Wang [16]	1	520	720 × 480	color	ped	90-min far-field intersection
2014	Saunier [11]	N/A	N/A	640 × 480	color	car,ped	1-intersection, 4-cameras, 3-months
2014	Saunier [11]	N/A	~ 1,000	800 × 600	color	car,ped	2-intersection, 1-cameras, 2-hours each
2014	Proposed Miovision	1,718	7,731,000	various	color	car	5-min video segments 19,244 training samples time-of-day, weather various road conditions 15 locations

which utilizes a parallelizable AdaBoost classifier described in Sec. III-C. A large-scale distributed computing system, described in Sec. III-D, is used to host the training and evaluation algorithms, allowing for efficient and parallelized processing. The resulting classification parameters can be distributed to live production systems once validated using the evaluation component.

A. Problem formulation

First, a general classifier, C is presented. C operates on a labeled sample, S_j , with known class correspondence, y_j , and with a corresponding feature vector, \vec{f}_j . Let D represent the entire population of traffic events, and S be the complete set of samples, a subset of D , used in training and validation. The classifier generates labels \hat{y}_j such that

$$\hat{y}_j = C(\vec{f}_j). \quad (1)$$

The ideal classifier minimizes the generalization error, ε , the difference between each \hat{y}_j and the actual value, y_j ,

$$\varepsilon = \sum_{j \in D} \begin{cases} 0, & \text{if } \hat{y}_j = y_j \\ 1, & \text{else.} \end{cases} \quad (2)$$

In practice, it is not possible to sample the entire population as implicitly indicated in (2). Only a sampled population, S , is available. S must be representative of D : a representative sample population is the cornerstone of positive and negative mining.

The parameters of C can then be estimated by minimizing ε using S instead of D . The discriminant function C can be represented using a simple complex function, or a set of weak functions or classifiers. In this paper, C is represented as discriminant function using a set of weak classifiers h , where each weak classifier h_c , contains a slope a_c , an offset b_c , and a threshold τ_c that segments a specified feature dimension d_c , into two regions, r_+ and r_- [54], where

$$r_+ = \begin{cases} 1, & \text{if } h_c \geq 0 \\ 0, & \text{else.} \end{cases}, \quad r_- = \begin{cases} 1, & \text{if } h_c < 0 \\ 0, & \text{else,} \end{cases} \quad (3)$$

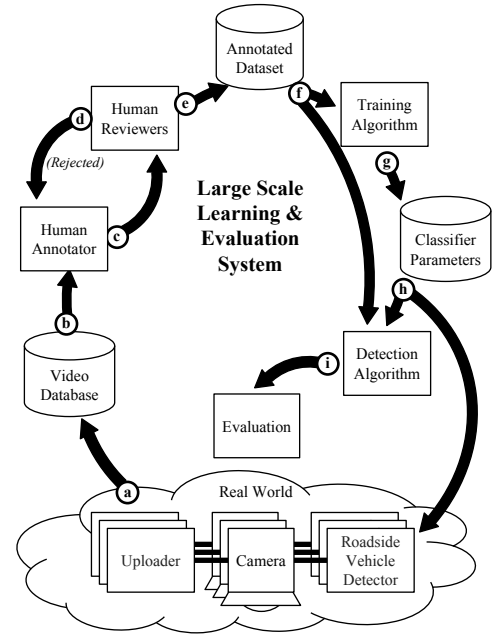


Fig. 4. Overview of the Large-Scale Machine Learning System: A network of cameras, each with a roadside processor, are deployed at signalized intersections. The processor records traffic video segments, which can be uploaded to the Video Database (a,b). Random still frames are sampled from the database and assigned to Human Image Observers (c), which annotate the frame to assign ground truth vehicle observations. All annotated observations are stored within the Training Samples Database (d). These samples are then used by the Training Algorithm (e) to generate Classifier Parameters (f). From the video segments (g) and the learned parameters (h) the Detection Algorithm generates vehicle observations (i). The same video segments are also annotated by a Human Video Observer to generate ground truth vehicle observations. The Evaluation step compares the detected (i) and annotated (j) observations to measure the effectiveness of the newly trained Detection Algorithm.

and

$$h_c = a_c H(f_{j,d_c} - \tau_c) + b_c \quad [54], \quad (4)$$

using H to represent the Heaviside step function. The complexity of the classifier depends on the number of weak classifiers, n_c , which should also be minimized for computational

efficiency. Using all of h_c , C is defined

$$C = \text{sign} \left(\sum_{c=1}^{n_c} h_c \right). \quad (5)$$

Given this formulation, there are two required steps in order to train a generalized real-time vehicle detector. First, a sampling process (Sec. III-B) must be established to obtain a representative S from D efficiently. Second, the classifier parameters, n_c and $\{a_c, b_c, \tau_c, d_c\} \forall c$, must be estimated using a distributed computing system (Sec. III-D).

B. Positive and negative mining

Data mining is required to select a representative S from D . The major steps are outlined in Alg. 1, and the data model is illustrated in Fig. 5. First, the mining begins with an initial manually dataset, S , of carefully chosen positive and negative samples from D . Then, the estimated parameters for C , that best classify vehicles and non-vehicles, are determined using S . The classifier is evaluated on a percentage, p , of random samples from D , and are compared to corresponding human annotations. Misclassified samples are then added to S and, using bias and variance analysis [41], noisy and overrepresented samples contained in S are removed. The process is repeated until the ε achieves a minimum, the C complexity, n_c , exceeds a threshold, or a maximum number of iterations is achieved.

Input: S manually curated initial sample of D with corresponding y_j

Output: S containing a representative sample of D

$$i \leftarrow 0;$$
while *isConverging*(ε, C, i) **do**
$$C \leftarrow \text{estimateClassifierParameters}(S, \{\dots, y_j, \dots\});$$
$$S^* \leftarrow \text{sparseRandomSampling}(D, p);$$

```
/* evaluate sample labels */
```

forall the j do
$$y_j^* \leftarrow \text{getManualLabel}(S_j^*);$$
$$\hat{y}_j^* \leftarrow C(S_i^*);$$

end

$$\varepsilon \leftarrow \text{calculateError}(\hat{y}_j^*, y_j^*);$$

```
/* add misclassified samples to
dataset */
```

foreach $\hat{y}_i^* \neq y_i^*$ **do**
$$S \leftarrow S \cup S_i^*;$$

end

$$S \leftarrow \text{rejectionSampling}(S);$$
$$i \leftarrow i + 1;$$
end**Algorithm 1:** Positive and negative mining

The data mining algorithm is the training algorithm represented in Fig. 4(f-g). The function, `estimateClassifierParameters(S)`, is implemented using AdaBoost, described in Sec. III-C and Fig. 7.

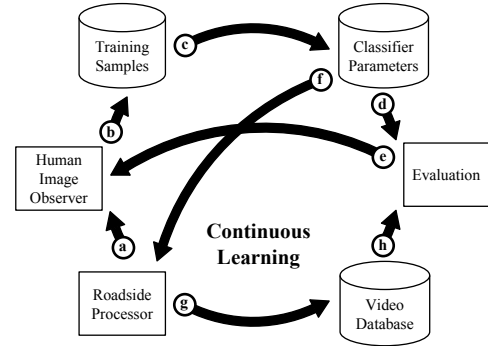


Fig. 5. Continuous Learning is a simple extension to the system whereby the Roadside Processor executes the current Detection Algorithm and randomly samples the video stream, indicating the presence or absence of vehicles. These samples are uploaded and assigned to a Human Image Observer (a), who independently annotates the ground truth and adds it to the Training Sample Database (b). Updated samples are used during Training (c) to produce new Classifier Parameters, which generate updated vehicle observations (d). Falsely classified observations are fed back to a Human Image Observer (e), who adds them to the Training Sample database. Should the new Classifier Parameters show improvements, they can be downloaded to the Roadside Processor in the field (f). Although much more costly, it is still possible to upload new video segments from the Roadside Processor (g) to continuously grow the testing set for Evaluation (h).

C. Classifier parameter estimation

Adaptive Boosting, or AdaBoost, is a well-known algorithm for efficiently building a single strong classifier, C , from a collection of weak classifiers, h . At each iteration, AdaBoost attempts to estimate $\{a_c, b_c, \tau_c, d_c\}$ for a single weak classifier, h_c , in this case one dimensional regression stumps, one for each feature dimension, that can best segment vehicles, \tilde{f}_p , from non-vehicles, \tilde{f}_n , with the minimal weighted classification error. Although initially each sample contributes equally when calculating classification error, samples with the greatest classification error are given more weight when computing error during subsequent iterations while samples with the least error are given less weight. Details of an AdaBoost implementation are provided by Friedman [44]. All parameters of C are determined once the algorithm converges on a minimal ε or maximum n_c . In the past, Sivaraman et al. implemented AdaBoost for a driver assistance program in 2013 [45]. The pseudo-code for a AdaBoost algorithm is shown in Alg. 2.

Features, \tilde{f}_j , used in this implementation are computed using the conventional Haar-like kernel. Each feature is calculated by multiplying a kernel, containing ones and negative ones, with pixel intensities from an image patch. However, when traversing all scales and translations of each Haar-kernel, the resulting feature-dimensionality becomes large. An $n_x \times n_y$ resolution image patch contains n_t possible kernel translations and n_s possible kernel scales, where n_t and n_s are both equal to $n_x \times n_y$; the kernel can be centered at any pixel and the kernel can have an area ranging from 1 to $n_x \times n_y$ pixels squared, see Fig. 6. The total number of possible features, n_f , for n_k potential kernels, is

$$n_f = n_t n_s n_k = (n_x n_u)(n_x n_u) n_k = n_x^2 n_u^2 n_k. \quad (6)$$

For a 42×42 resolution image patch with eight unique

Input: (S, \vec{y})
Output: h
 $h \leftarrow \{\}, n_j \leftarrow \sum_{\forall j} 1, c \leftarrow 0;$
 $w_{j,0} = \frac{1}{n_j} \forall j;$
 $\vec{f}_j \leftarrow \text{extractFeatures}(S_j) \forall j;$
 $n_f \leftarrow \text{numFeatureDimensions}();$
 $e \leftarrow 0 \cdot [1 \dots n_f];$
while $\text{isConverging}(c, e)$ **do**
 /* update weights */
 forall the j **do**
 $w_{j,c} \leftarrow \exp(-y_j C(\vec{f}_j))$
 end
 /* calc candidate classifiers */
 $e \leftarrow 0;$
 for $i = 1 : n_f$ **do**
 $d_c \leftarrow i;$
 /* adaboostEst implements [44], [54] */
 $\{a_c, b_c, \tau_c, d_c, e_i, w^i\} = \text{adaboostEst}(\vec{f}, \vec{y}, w, c);$
 $h_{ci} \leftarrow \{a_c, b_c, \tau_c, d_c, e_i\};$
 end
 $w \leftarrow w^{i*}, h_c \leftarrow h_{ci*}, \text{ such that } i^* = \arg \min_i e_i;$
 $h \leftarrow h \cup h_c;$
 $c \leftarrow c + 1;$
end

Algorithm 2: AdaBoost iteration

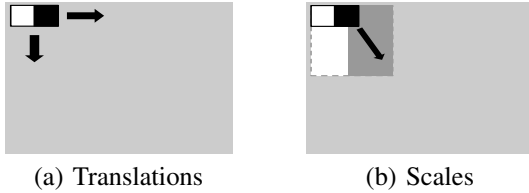


Fig. 6. An example Haar kernel on an image patch (gray). There is a feature value associated with (a) all translations and (b) all scales for each translation.

kernels, $n_f = 24,893,568$, neglecting boundary conditions for simplification.

To account for such a high-dimensional feature space, the proposed learning system, shown in Fig. 7, integrates a parallelized AdaBoost implementation to estimate an optimal set of parameters. The following AdaBoost components,

- 1) Haar-like features, \vec{f}_j ,
- 2) sample weights, w ,
- 3) and weak-classifier candidates, h_{ci}

can be calculated independently and are processed in parallel. A distributed computing system can implement these steps through parallel processing and distributed memory.

D. Distributed and scalable computing

Due to the iterative nature of algorithm design, the training and evaluation system executes many times during the course of development as parameters are tuned and as new algorithm ideas are integrated. As researchers propose changes, feedback is necessary to determine if their changes improve accuracy

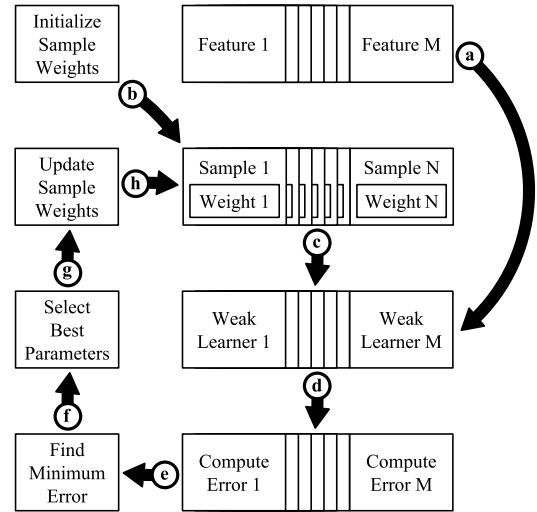


Fig. 7. First, for this distributed learning system, the Haar like features are extracted and sent to an iterative AdaBoost learning system, where multiple processes compute weak learners corresponding to a single feature. Then a master machine combines all the learner to select a best learner. Further, the best parameters are used to update the sample weights for the next iteration.

on desired real-world scenarios. Evaluating the dataset in a reasonable amount of time is critical to a practical development process. This section outlines the requirements, workflow, and tools for a distributed computing environment capable of efficient processing.

The evaluation system must be robust to errors occurring during an evaluation; the system must provide feedback and fail gracefully, especially if costs are incurred if the evaluation system continuous to produce erroneous results. Consecutive runs of the same algorithm on the same dataset should be deterministic and produce the same results and recover from unexpected network issues or individual processor failures. The evaluation system must also be easy to use so that researchers can focus on video detectors and do not need to worry about the underlying evaluation infrastructure. The system must be able to accommodate additional data added or removed from the annotation dataset, and must allow evaluation on subsets of the dataset. The accuracy and performance results of the candidate vehicle detection algorithm must then be reported and stored for future reference.

The distributed system workflow is an implementation of the general MapReduce pattern [55].

- 1) Select the complete or a query defining a subset of the annotated dataset.
- 2) Upload or specify a vehicle detection algorithm.
- 3) Select an appropriate set of trained parameters.
- 4) An evaluation task is created for each localization annotation from the dataset specified above.
- 5) The evaluation tasks are dispatched to processors.
- 6) Individual processors obtain required data and execute the evaluation task.
- 7) Accuracy and performance results are aggregated.
- 8) Statistics are calculated from the aggregation.
- 9) Results are reported and achieved for future comparison.

The task creation, dispatch, processing, and aggregation steps

are executed on numerous Amazon EC2 [50] instances using Amazon S3 [50] for annotation and video data storage. Large amounts of video can be shared between EC2 instances and S3 using internal Amazon high speed networks. By default the number of EC2 instances is limited to 20 simultaneous instances per instance type, but can be increased if needed, and each instance currently contains up to 32 cores. The EC2 solution has sufficient computing power for the current dataset needs.

IV. EXPERIMENTAL VALIDATION

A. Data acquisition

Data collected for this study is obtained from prototype Miovision Permanent Connected Intersection Count Stations, distributed over 15 locations. While GIS locations are recorded for each intersection, such information is anonymized, but access to time of day information and a broad provincial context is available to researchers. Stations operate at the roadside in all environments, day and night. Videos encompass the entire intersection because stations are equipped with fisheye lenses. The output video is encoded to H.264 with a resolution of 1536×1536 at 15 frames per second. The processor then rectifies regions of video into perspective views that contain sets of adjacent lanes.

In 2014, data obtained from the portable Miovision Scout Video Collection Units [56] consist of approximately 52,000 unique North American locations, intersections and roadways, with an additional 9,000 across Europe. A subset of this data is currently being added to the proposed traffic event dataset and the authors intend to integrate portions of 2015 data as it is collected.

B. Video dataset

The ITS traffic event dataset currently consists of over 7.7 million video frames and continues to expand monthly. Each frame can be annotated in several ways. Configuration defines vehicle detection zones relative to the roadway, see Fig. 8(a). Depending on the study type, this zone may encompass the entire visible lane or a region that is only large enough to fit one vehicle. The annotators also have the ability to label environmental conditions associated with the video or individual frames. The vehicles can be localized allowing the annotators to specify when a vehicle is present at one or more locations, as illustrated in Fig. 8(b) with red and blue regions. The boundaries for objects in the scene can also be drawn by annotators, see Fig. 8(c). For each case, the annotator can label objects with appropriate classes, e.g. passenger car. Once annotation is complete, a review process ensures that the annotations are correct; a percentage of video frames are annotated by multiple annotators to provide a measure of annotator variability and, similarly, multiple reviews provide reviewer variability metrics. The review process is continuously improved through training to reduce the probability of accepting a misannotated object.

The proposed video dataset is generated using the process detailed in Sec. III-B. Initially, a random sample of the acquired video data is obtained. Each video is configured,

and for each video frame, every vehicle is localized through human annotation. These localized vehicles become the testing dataset. A subset of video frames are initially randomly sampled from the testing dataset to become the training dataset. Boundaries for each vehicle are annotated for each sample in the training dataset. Using the initial training dataset, the classifier parameters were trained and the results were evaluated against the testing set. Testing samples with poor performance are then given to human annotators for boundary annotation before being moved into the training dataset. Meanwhile, the testing set continues to grow as newly acquired video is randomly sampled and added. The composition of the resulting dataset is detailed in the following sub-section, Sec. IV-C. The proposed dataset contains about 7.7 million samples of localization and 19,244 samples of boundary annotations.

C. Proposed large-scale ITS dataset

The proposed dataset contains 3,082 five minute video segments, a total of 256.8 hours, acquired from 15 locations throughout Canada. Since there are a total of 182 distinct lanes, each lane is sampled, on average, for about 1.4 hours, with a corresponding 16.9 discontinuous segments of continuous five minute video per lane. The centroid of each vehicle is specified at least once per detection zone. The dataset contains approximately 209 hours recorded during the day, 121 hours at night, and 51.3 hours at dusk or dawn. The dataset contains approximately 4.5 hours of light rain, 28.4 hours of heavier rain, 7.5 hours of light snow, 25.5 hours of regular snow, and 7.5 hours of heavy snow. There are also 187.6 hours of clean roadways, 8 hours with some snow, 28.9 hours of snow covered, and 61.2 of wet roads.

Fig. 9 illustrates several examples of various conditions represented in the dataset. Sec. IV presents detailed breakdown of detector accuracy when evaluated against additional sets of conditions, Fig. 10. Each video frame contains vehicles with resolution between 14×14 and about 80×80 pixels, with median vehicles resolution as 26×26 pixels. The video data has been acquired over a span of 2 years, with a range of weather and operating conditions.

D. Vehicle event detection system

As described in a previous work [39], the trained classifier parameters are incorporated into a vehicle event detection system, which utilizes background modelling, time of day estimation, object detection, Kalman filter based tracking, and AdaBoost. Each intersection is manually configured with view specific metadata including a collection of incoming and outgoing lanes or zones. The vehicle classifier is combined with background subtraction and an online updated parametric lane model to associate vehicles to lanes and zones for vehicle presence detection. The classifier parameters are trained using the proposed ITS dataset and continuous learning process.

E. Evaluation metrics

Video detector accuracy is reported based on the four detector modes defined in the NEMA TS-2 standard [57].

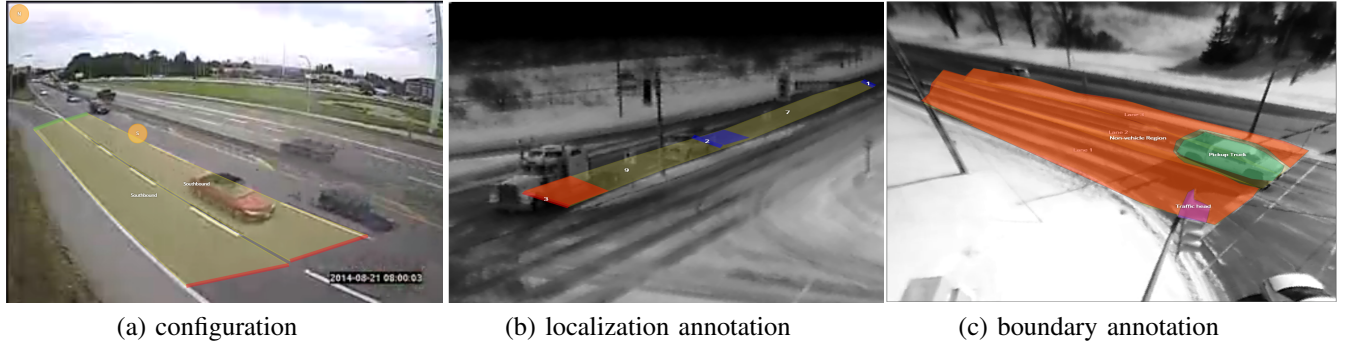


Fig. 8. (a) Two southbound vehicle presence detection zones are annotated as yellow polygons. Vehicles enter each zone at the green line segment and exit at the red line. (b) Three vehicle regions define the lane entry, mid-point and exit. A truck has just reached the exit region and is labelled as a large truck with trailer. (c) Boundaries are drawn around objects of interest, which are also classified, e.g. road, pick-up truck, or signal head.



Fig. 9. These figures represent several of many conditions, resolutions, camera perspectives, and locations contained in the proposed ITS dataset.

- 1) Pulse: a pulse of duration 100 to 150 *ms* that is triggered when a vehicle enters the detection zone.
- 2) Controlled output: identical to pulse, but with a configurable pulse duration.
- 3) Continuous presence: a signal is generated for as long as a vehicle is present in the detection zone.
- 4) Limited presence: identical to continuous presence, but with a configurable maximum duration. Note that the signal may end before the maximum duration if the detected vehicle leaves the detection zone early

Throughout this paper, presence mode is used exclusively because the authors have focused on detection applications requiring this metric. Other applications and systems require some or all of these operational modes.

The evaluation system reports a confusion matrix representing the following items.

- 1) True positive, *TP*, (true call): a vehicle is present and a corresponding detection call is correct.

- 2) False negative, *FN*, (false call): a vehicle is not present in the detection zone, despite a video detection. Detectors may fail in this way if a vehicle was previously in the detection zone, but did not detect the vehicle leaving the zone, also known as a ‘stuck on call’[6].
- 3) False positive, *FP*, (missed call): a vehicle is present in the detection zone, but is not detected. This failure may occur if the detector initially identifies the vehicle, but fails to continuously detect the vehicle the entire time it is in the zone, also known as a ‘dropped call’[6].
- 4) True negative, *TN*, (true non-call) a vehicle is not present in the detection zone, which the detector correctly reports.

The overall accuracy, α , is derived from the confusion matrix,

$$\alpha = \left(\frac{TP + TN}{TP + FN + FP + TN} \right). \quad (7)$$

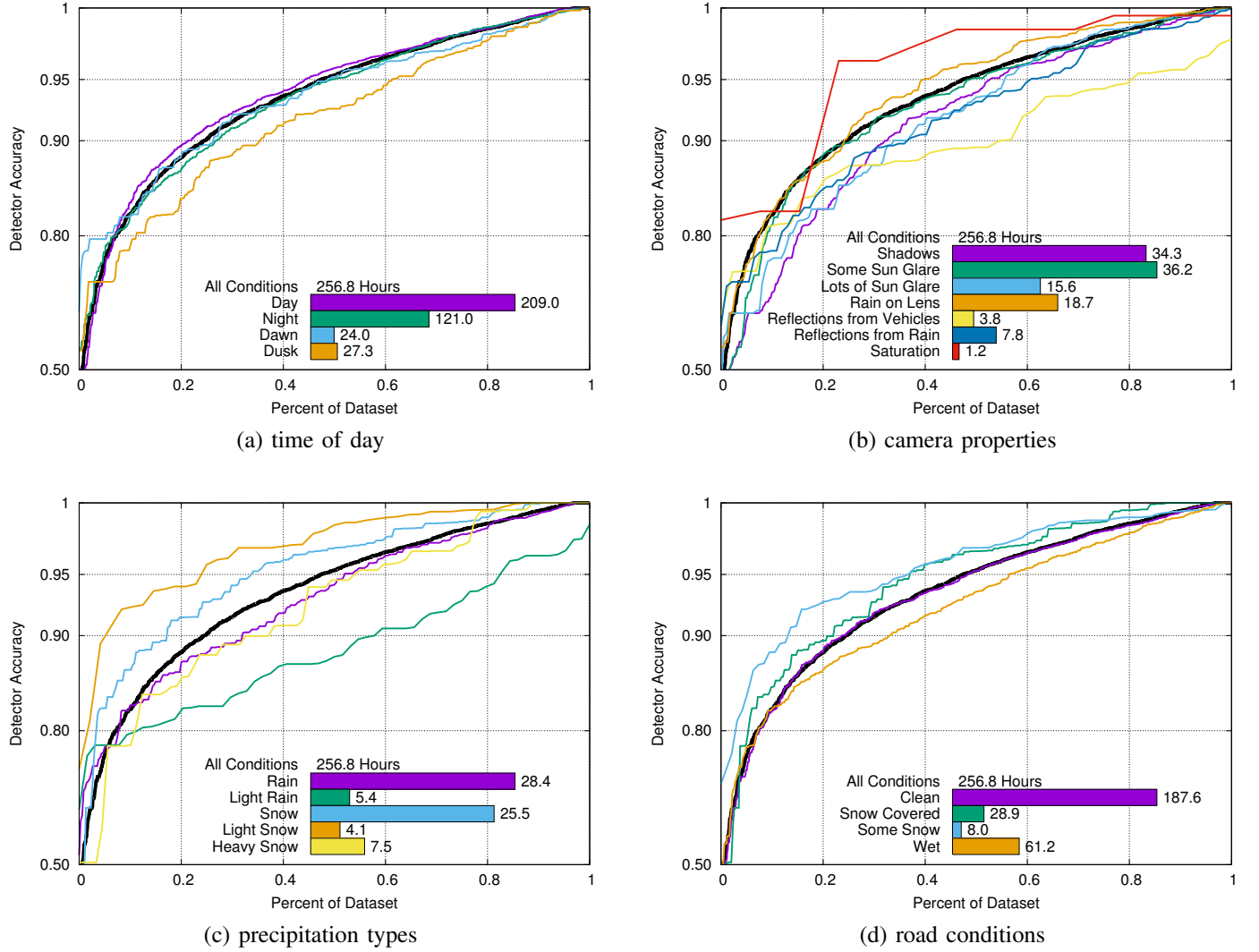


Fig. 10. Cumulative distributions of four annotation categories. From time of day (a), the trained video detector performs better at dawn than at dusk (27.3 hours of video). The authors' video detector detects vehicles through headlights during the night, and performs better at dawn because it was observed that drivers typically have headlights on at dawn more often than they have headlights on at dusk. From observing (b) camera properties and (c) precipitation types, the detector performs worse in light rain conditions and also when reflections from vehicles are present. Also from observing (c) precipitation types and (d) road conditions, it should be noted that the video detector performs very well in snow and snow covered conditions, which may be due to the higher contrast between vehicles and their surrounding.

In addition to the metrics above, the evaluation system also measures runtime, and receiver operating characteristic (ROC) curves related to customer accepted ratio of true positives compared to false positives. Runtime is a useful metric particularly if a real-time detection algorithm is being evaluated.

F. Overall evaluation accuracy

The overall accuracy is evaluated on the testing dataset described in Sec. IV-C. The vehicle detection accuracy for each video is sorted from lowest to highest is 1/2 of the dataset have a vehicle presence accuracy in excess of 95% and that 19/20 of the dataset have an accuracy in excess of 78%. In addition, for 1/2 of the dataset, counting metrics exceed 78% accuracy.

G. Dataset composition vs. accuracy

Fig. 10 illustrates the accuracy of the authors' video based vehicle detector using the proposed ITS dataset for training and testing. The overall detection accuracy is illustrated, the solid black curve exceeds 95% for 1/2 and 78% for 19/20 of the dataset. Using the annotation labels, accuracy for the video detector in a variety of conditions can be calculated. The four sub-figures in Fig. 10 present comparisons of time of day, environmental conditions, precipitation types, and road conditions. From these figures, researchers can identify scenarios with the lowest accuracy and improve the detection accuracy by modifying and evaluating proposed video detector designs and by improving the training process by incorporating more training samples into the dataset for poorly performing scenarios. There are additional annotation labels that are collected but not illustrated below, such as traffic density, distance of the lane from the camera, number of adjacent incoming or

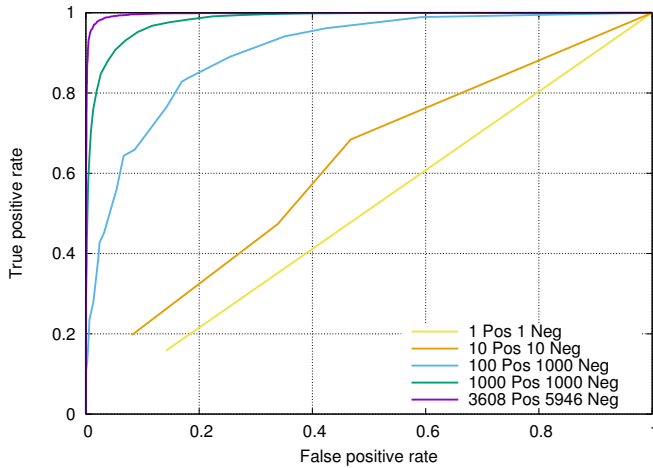


Fig. 11. ROC curve illustrating that the detection classifier achieves the maximum true positives and minimum false positives.

outgoing lanes, and distributions of vehicle classes.

H. Sample size vs. evaluation accuracy

Sample quantity and diversity play a significant role in designing a real-time and continuous learning based vehicle detection system. A positive and negative mining technique is applied to collect a diverse training dataset. The receiver operating characteristic (ROC) curve, illustrated in Fig. 11, indicates that the detector accuracy, evaluated on the testing set, increases as the quantity of training samples increases. The primary reason for this phenomenon is that a reasonable quantity of training samples is required to establish sufficient diversity to represent the testing dataset. Further, Fig. 11 indicates that the designed AdaBoost based strong classifier uses a threshold of 0.058301, similar to the ideal value of 0, to maximize true positives and minimize false positives when deployed in field.

V. DISCUSSION AND CONCLUSIONS

This paper presents an ITS dataset for the purpose of real-time vehicle detection. The proposed positive and negative mining process allows the creation of an ITS dataset by selecting training and testing samples that are representative of the real-world. The process also culls the dataset by removing noisy and redundant samples. As shown, a detector can be trained using significantly fewer, but representative, samples than using an entire dataset. Positive and negative mining avoided the need to annotate 7.7 million video frames containing vehicle boundaries and allowed the detector to be trained on only 19,244 boundaries instead. The detector achieved 95% accuracy for 1/2 and over 78% accuracy for 19/20 of the data, evaluated on all 256.8 hours of the video, containing 7.7 million video frames of localized vehicles; localization is significantly more efficient than boundary annotation.

This shows the effectiveness of continuous learning applied to real-time vehicle detection. The continuous learning process utilizes positive and negative mining to efficiently represent

a diverse and compact training dataset from a massive large-scale population of roadways and operating environments. The process has generated, to the best of the authors' knowledge, the largest and most diverse ITS dataset to date. Further, the continuously trained detector parameters, using a large-scale distributed computing system, are transmitted and incorporated into real-world video detectors as the ITS dataset continues to update. The learning process and training system allow researchers to quickly evaluate new algorithms, not only for detection, but for future ITS applications; the researcher can focus on algorithm design instead of data management.

The work presented here is only the beginning. The authors plan to expand the scope of the ITS dataset by including data acquired from all over the world and to start annotating additional ITS classes, such as pedestrians, bicycles, buses, and various classes of trucks. The authors also fully intend to provide API access to this dataset, allowing the computer vision researcher community to evaluate and train their own algorithms on a large-scale. As for the demonstrated video detector, the immediate goal is to improve accuracy for rainy and highly reflective conditions.

REFERENCES

- [1] J. Zhang, F.-Y. Wang, K. Wang, and X. X. Wei-Hua Lin, "Data-driven intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec 2011.
- [2] A. Ambardekar, M. Nicolescu, G. Bebis, and M. Nicolescu, "Visual traffic surveillance framework: classification to event detection," *J. of Electron. Imaging*, vol. 22, no. 4, pp. 041 112–041 112, Aug 2013.
- [3] D. Desai and S. Somani, "Instinctive traffic control and vehicle detection techniques," *Int. J. of Scientific & Eng. Research*, vol. 5, no. 1, pp. 2192–2195, Jan 2014.
- [4] U. Lee and M. Gerla, "A survey of urban vehicular sensing platforms," *Computer Networks*, vol. 54, no. 4, pp. 527–544, 2012.
- [5] L. Tong and Z. Li, "Study on the road traffic survey system based on micro-ferromagnetic induction coil sensor," *Sensors & Transducers*, vol. 170, no. 5, pp. 73–79, May 2014.
- [6] J. C. Medina, R. F. Benekohal, and M. V. Chitturi, "Evaluation of video detection systems volume 1-effects of configuration changes in the performance of video detection systems," Illinois Center for Transportation, Tech. Rep. ICT-08-024, 2008.
- [7] B. Morris and M. Trivedi, "Robust classification and tracking of vehicles in traffic video streams," in *IEEE Conf. Intell. Transp. Syst.*, Sept 2006, pp. 1078–1083.
- [8] S. Sivaraman and M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 267–276, June 2010.
- [9] —, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec 2013.
- [10] P. A. Viola and M. J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 1, June 2001, pp. 511–518.
- [11] N. Saunier, H. Ardo, J.-P. Jodoin, A. Laureshyn, M. Nilsson, A. Svensson, L. Miranda-Moreno, G.-A. Bilodeau, and K. Astrom, "A public video dataset for road transportation applications," in *Transp. Res. Board Annu. Meeting Compendium of Papers*, 2014, pp. 14–2379.
- [12] P. S. Li, I. E. Givoni, and B. J. Frey, "Learning better image representations using flobjct analysis," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2011, pp. 2721–2728.
- [13] X. Wang, X. Ma, and W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 539–555, Jan 2009.
- [14] N. Saunier, T. Sayed, and C. Lim, "Probabilistic collision prediction for vision-based automated road safety analysis," in *IEEE Conf. Intell. Transp. Syst.*, 2007, pp. 872–878.

- [15] C. Papageorgiou, T. Poggio, M. Oren, P. Sinha, and E. Osuna. (2000) Cbcl car database #1 @ONLINE. [Online]. Available: <http://cbcl.mit.edu/software-datasets/CarData.html>
- [16] M. Wang and X. Wang, "Automatic adaptation of a generic pedestrian detector to a specific traffic scene," in *IEEE Conf. Comput. Vis. Pattern Recog.*, Colorado Springs, CO, June 2011, pp. 3401–3408.
- [17] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1475–1490, Nov 2004.
- [18] H. Schneiderman and T. Kanade, "A statistical method for 3d object detection applied to faces and cars," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2000, pp. 746–751 vol.1.
- [19] D. Makris. (2001) Index of /pub/pets2001/ @ONLINE. [Online]. Available: <ftp://ftp.pets.rdg.ac.uk/pub/PETS2001/>
- [20] S. M. Bileschi, "Streetscenes: Towards scene understanding in still images," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, May 2006.
- [21] J. C. Medina, R. F. Benekohal, and M. V. Chitturi, "Evaluation of video detection systems volume 4-effects of adverse weather conditions in the performance of video detection systems," Illinois Center for Transportation, Tech. Rep. ICT-09-039, 2009.
- [22] P. Siva, C. Russell, and T. Xiang, "In defence of negative mining for annotating weakly labelled data," in *ECCV (3)*, ser. Lecture Notes in Computer Science, vol. 7574. Springer, 2012, pp. 594–608.
- [23] H. Sanson, L. Loyola, and D. Pereira, "Scalable distributed architecture for media transcoding," in *Algorithms and Architectures for Parallel Processing*, ser. Lecture Notes in Comput. Sci., Y. Xiang, I. Stojmenovic, B. Apduhan, G. Wang, K. Nakano, and A. Zomaya, Eds., vol. 7439. Springer Berlin Heidelberg, 2012, pp. 288–302.
- [24] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, "Large scale distributed deep networks," in *NIPS*, 2012.
- [25] K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. J. Wright, "Performance analysis of high performance computing applications on the Amazon Web Services cloud," in *IEEE Int. Conf. on Cloud Computing Technology and Science*, 2010, pp. 159–168.
- [26] L. A. Klein, *Traffic Detector Handbook*, 3rd ed., ser. 10. Rancho Palos Verdes, CA: Federal Highway Administration, 2006, vol. 1.
- [27] W. Xiang, C. Otto, and P. Wen, "Automated vehicle classification system using advanced noise reduction technology," in *Int. Conf. on Signal Process. and Commun. Syst.*, 2007, pp. 17–19.
- [28] W. Birk, J. Eliasson, P. Lindgren, E. Osipov, and L. Riliskis, "Road surface networks technology enablers for enhanced ITS," in *Veh. Networking Conf.*, 2010, pp. 152–159.
- [29] P. Handel, J. Ohlsson, M. Ohlsson, I. Skog, and E. Nygren, "Smartphone based measurement systems for road vehicle traffic monitoring and usage based insurance," *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–11, Dec 2013.
- [30] M. R. Friesen and R. D. McLeod, "Bluetooth in intelligent transportation systems A survey," *Int. J. of Intell. Transp. Syst. Research*, p. 1, May 2014.
- [31] M. Song, D. Tao, and S. J. Maybank, "Sparse camera network for visual surveillance – A comprehensive survey," *CoRR*, vol. abs/1302.0446, pp. 1–41, Feb 2013.
- [32] R. Salakhutdinov, J. B. Tenenbaum, and A. Torralba, "Learning with hierarchical-deep models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1958–1971, 2013.
- [33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2005, pp. 886–893 vol. 1.
- [34] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.
- [35] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008.
- [36] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Int. Conf. on Comput. Vis.*, Nov 2011, pp. 2564–2571.
- [37] X. Wang, X. Ding, and C. Liu, "Gabor filters-based feature extraction for character recognition," *Pattern Recog.*, vol. 38, no. 3, pp. 369–379, Mar. 2005.
- [38] S. Krig, *Computer Vision Metrics: Survey, Taxonomy, and Analysis*, 1st ed. Berkely, CA, USA: Apress, 2014.
- [39] J. Eichel, A. Mishra, N. Miller, N. Jankovic, and K. McBride, "Decoding the spatiotemporal scene of a road-traffic intersection for real time event detection," in *IEEE Conf. Comput. Vis. Pattern Recog. Scene Understanding Workshop*, Columbus, OH, June 2014.
- [40] L. Fan and K. Poh, "A comparative study of PCA, ICA and Class-Conditional ICA for naive bayes classifier," in *Computational and Ambient Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4507, pp. 16–22.
- [41] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. of Comput. and Syst. Sci.*, vol. 55, no. 1, pp. 119 – 139, 1997.
- [42] R. Harrison and C. Koch, "A robust analog VLSI reichardt motion sensor," *Analog Integrated Circuits and Signal Processing*, vol. 24, no. 3, pp. 213–229, 2000.
- [43] J. P. H. van Santen and G. Sperling, "Elaborated reichardt detectors," *J. Opt. Soc. Am. A*, vol. 2, no. 2, pp. 300–321, Feb 1985.
- [44] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 04 2000.
- [45] S. Sivaraman and M. Trivedi, "Integrated lane and vehicle detection, localization, and tracking: A synergistic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 906–917, June 2013.
- [46] P. M. Long and R. A. Servedio, "Random classification noise defeats all convex potential boosters," *Machine Learning*, vol. 78, pp. 608–615, 2008.
- [47] R. L. Moore, D. L. Hart, W. Pfeiffer, M. Tatineni, K. Yoshimoto, and W. S. Young, "Trestles: A high-productivity HPC system targeted to modest-scale and gateway users," in *TeraGrid Conf.: Extreme Digital Discovery*, 2011, pp. 25:1–25:7.
- [48] P. M. Papadopoulos, C. A. Papadopoulos, M. J. Katz, W. J. Link, and G. Bruno, "Configuring large high-performance clusters at lightspeed: A case study," *Int. J. of High Performance Computing Applicat.*, vol. 18, no. 3, pp. 317–326, May 2004.
- [49] F. Jokhio, A. Ashraf, S. Lafond, I. Porres, and J. Lilius, "Prediction-based dynamic resource allocation for video transcoding in cloud computing," in *Euromicro Int. Conf. on Parallel, Distributed and Network-Based Processing*, Feb 2013, pp. 254–261.
- [50] J. Murty, *Programming Amazon Web Services - S3, EC2, SQS, FPS, and SimpleDB*. O'Reilly, 2008.
- [51] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 319–336, Feb 2009.
- [52] B. Tamersoy and J. Aggarwal, "Robust vehicle detection for tracking in highway surveillance videos using unsupervised learning," in *AVSS*, Sept 2009, pp. 529–534.
- [53] P. Roth and H. Bischof, "Active sampling via tracking," in *CVPRW*, June 2008, pp. 1–8.
- [54] A. Torralba, K. P. Murphy, and W. T. Freeman, "Sharing features: efficient boosting procedures for multiclass object detection," in *IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 2, June 2004, pp. 762–769.
- [55] J. Ekanayake, S. Pallickara, and G. Fox, "Mapreduce for data intensive scientific analyses," in *eScience, IEEE Fourth Int. Conf*, Dec 2008, pp. 277–284.
- [56] K. McBride, C. Fairles, K. Madill, D. Zhang, T. Brijpaul, D. Thompson, and V. Silagadze, "Method and system for analyzing multimedia content," Patent US 8 204 955, Jun 19, 2012.
- [57] *Traffic Controller Assemblies with NTCIP Requirements*, NEMA Standards Publication TS 2-2003 (R2008), National Electrical Manufacturers Association Std., Rev. 02.06, 2003.



Justin A. Eichel computer vision architect at Miovision Technologies. PhD from Vision and Image Processing Lab, Systems Design Engineering, University of Waterloo. Previous experience as self-employed consultant related to multi-spectrum tracking and medical image processing researcher. Justin is skilled at statistical modelling, pattern recognition, and machine learning.



Mohan A. Thomas software developer at Miovision Technologies and has obtained his Bachelors of Applied Science from Systems Design Engineering at the University of Waterloo. Mohan is experienced with traffic engineering and systems integration, and has previous experience at Nuance Communications Inc. and National Research Council Canada.



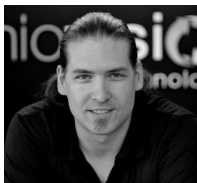
Akshaya Mishra applied research scientist at Miovision Technologies and concurrently holds an Adjunct faculty position at the University of Waterloo. He holds a PhD from the University of Waterloo, as well as an M. Tech and B.E. in Electrical Engineering from the Indian Institute of Technology. Akshaya focuses in statistical modelling, image processing, and pattern recognition.



Nicholas Miller applied research scientist at Miovision Technologies having received his master's and bachelor's of Mathematics from the School of Computer Science at the University of Waterloo. He has studied visual perception of events and physical interactions and is experienced in video sensing, distributed computing systems, machine learning, and optimization.



Doug Swanson VP Engineering at Miovision Technologies. Responsible for the MioLabs Research team developing computer vision, traffic simulation and optimization technologies and has previous experience at Blackberry, Cisco and Nortel. Doug obtained a Bachelor of Applied Science in System Design Engineering from the University of Waterloo.



Nicholas Jankovic embedded developer at Miovision Technologies and is a licensed Professional Engineer of Ontario and holds a Master of Engineering Science in Sensing and Mechatronic Systems from the University of Western Ontario. He is experienced in machine vision, has designed camera systems, and embedded software development and verification.



Joel Keller an embedded software specialist at Miovision Technologies, experienced in firmware development, hardware/software co-design, and software architecture. Joel holds a Bachelor's of Mathematics in Computer Science from the University of Waterloo.



Tyler Abbott software developer at Miovision Technologies and has received an Ontario College Advanced Diploma in Computer Science Technology with High Honours from Sheridan College. Tyler has previously worked designing software at Blackberry and has experience with software development, rapid prototyping, and data visualization.